

Optimization of ID3 Structure for Academic Performance Analysis using Ant Colony Algorithm

Dedin Fathudin^{1,*}, Erlin Windia Ambarsari², Aulia Paramita²

¹ Faculty of Computer Science, Informatics Engineering, Universitas Pamulang, South Tangerang, Indonesia

² Faculty of Engineering and Computer Science, Informatics Engineering, Universitas Indraprasta PGRI, Jakarta, Indonesia

Email: ^{1,*}dosen00398@unpam.ac.id, ²erlinunindra@gmail.com, ³aulia.pps@gmail.com

Correspondence Author Email: dosen00398@unpam.ac.id

Submitted: 16/06/2024; Accepted: 25/06/2024; Published: 26/06/2024

Abstract—This study investigates the optimization of the ID3 algorithm for academic performance analysis using the Ant Colony Optimization (ACO) method. The primary research problem addressed is the inefficiency and overfitting of traditional ID3 in complex and noisy datasets. Therefore, the ACO method is integrated to enhance the ID3 structure, improving classification accuracy and computational efficiency. The research objectives include developing a decision tree model based on assignment, mid-term, and final exam scores for student performance evaluation. The method combines ID3's decision-making capabilities with ACO's optimization process, which uses pheromone trails to find optimal paths in constructing the decision tree. Temporary results show that the ACO-ID3 model achieves an accuracy of 85% with improved consistency and lower variability compared to the traditional ID3 model, which has an accuracy of 89% but higher variability; this indicates that while traditional ID3 may slightly outperform in accuracy, the ACO-ID3 model provides more stable and reliable performance across different data subsets. The study concludes that ACO-ID3 is a practical and effective tool for academic performance analysis, particularly in cases requiring consistent and reliable classification.

Keywords: ID3 algorithm; Ant Colony Optimization; Academic Performance Analysis; Decision Tree; Classification Accuracy

1. INTRODUCTION

The 'If' condition is a fundamental tool for formulating statements in software engineering, particularly programming. The 'If' condition is indispensable in programming as it provides the foundational mechanism for conditional execution, enabling programs to make decisions, handle errors, and execute repetitive tasks efficiently. However, this can backfire when nested 'If' statements are used excessively and become overly complex, as it can significantly slow down data processing within the program and, in some cases, cause the software to hang. This problem of complexity and inefficiency in decision-making forms the basis of our research, where we propose a novel solution using the ACO-ID3 model for academic performance analysis.

Therefore, one solution to this problem is implementing the Decision Tree method, which can classify each condition based on specific criteria. The Decision Tree method plays a significant role in programming by providing a structured approach to decision-making. It simplifies complex conditional logic by breaking down decisions into a tree-like model of choices, where each node represents a condition, and each branch represents the outcome of that condition. This structure helps organize and visualize the decision process, making it easier to analyze. In practice, Decision Trees can enhance the efficiency and maintainability of programs. Streamlining the decision paths reduces the risk of performance issues associated with deeply nested 'If' statements. However, the Decision Tree had other problems, such as overfitting, instability, bias, and short-sighted decisions.

Several studies have implemented Decision Tree approaches, including ID3 (Iterative Dichotomiser 3) [1]–[4], C4.5 [5]–[9], and CART (Classification and Regression Trees) [10]–[14]. In this study, we focus on developing ID3 as a straightforward method. The implementation of ID3 was conducted to build 'If' statements in the academic assessment of students at the Universitas Indraprasta PGRI for the Programming 3 course as a sample, where typically, the assessment uses weighted scores. However, in this study, the weighting was removed, and only the scores from assignments, midterm exams, and final exams were used to construct the decision tree structure with the ID3 approach. The results will then determine whether the outcomes of the ID3 approach will be the same as those from the weighted scoring system.

However, several weaknesses emerge when the ID3 Decision Tree becomes more complex. One major drawback of ID3 is its tendency to overfit the data, especially when dealing with noisy or incomplete datasets, particularly in Traditional ID3 [15]. While the tree may perform well on training data, it may not generalize well to missing data, resulting in poor predictive performance. Additionally, ID3 can produce more branches, especially with continuous data, making the Model challenging to interpret and manage. This complexity can lead to inefficiencies in computation and storage, making it less practical for large-scale applications. Therefore, ID3 needs to add other techniques to optimize the tree structure, such as looking for the shortest branch paths and building 'If' statements efficiently, which is necessary.

With this in mind, this study examines the Ant Colony Optimization method (ACO) to support ID3 in constructing decision trees for Academic Performance Analysis. The ant colony algorithm, a distributed problem-solving technology, is widely used in optimization problems due to its efficiency and effectiveness in handling complex scenarios with many objects or tasks. This algorithm, inspired by the behavior of ants, utilizes pheromone communication to create solutions through interactions among artificial ants, resembling the pathfinding behavior of

real ants [16]–[18]. By dividing problems into smaller parts and assigning them to individual workers, the algorithm efficiently finds combined solutions to form an optimal outcome [18], [19]. Additionally, in distributed constraint solving, the ant colony optimization approach enhances bidirectional constraint solving, ensures agent privacy, and improves local cost estimation mechanisms, outperforming incomplete algorithms in distributed environments [20]. ACO has been successfully applied to routing in telecommunication networks, such as the Traveling Salesman Problem (TSP), demonstrating its effectiveness in finding optimal solutions [21]. Although several studies have combined the ACO algorithm with other methods [22], [23], no research has employed the ACO-ID3 approach directly for this case. Therefore, this study aims to explore how the Ant Colony Optimization (ACO) method can enhance the performance of the ID3 algorithm in academic performance analysis, particularly in addressing issues of overfitting and computational efficiency.

2. RESEARCH METHODOLOGY

2.1 ID3 (Iterative Dichotomiser 3)

ID3 is a decision tree algorithm introduced by Ross Quinlan. It is also known as Traditional or Classical ID3. This algorithm generates a decision tree from a dataset using entropy and information gain [24]. Generally, the pseudocode in the ID3 algorithm is as follows:

Input: Dataset S, Attributes A, Target attribute T

Output: Decision tree

1. function ID3(S, A, T):
2. Create a root node for the tree
3. if all instances in S have the same label, return the single-node tree Root with that label
4. if A is empty, return the single-node tree Root with label = most common value of T in S
5. else:
6. A_best = Attribute with highest Information Gain from A
7. Root = Create a decision node that splits on A_best
8. for each value v_i of A_best:
9. S_i = {instances in S with A_best = v_i}
10. if S_i is empty:
11. Add a leaf node with label = most common value of T in S
12. else:
13. Add the subtree ID3(S_i, A - {A_best}, T) to the corresponding branch of Root
14. return Root
- // Calculate Entropy
15. function Entropy(S):
16. $H(S) = - \sum (p_i * \log_2(p_i))$ for each unique value i in T
17. return H(S)
- // Calculate Information Gain
18. function InformationGain(S, A):
19. $IG(S, A) = Entropy(S) - \sum (|S_v| / |S| * Entropy(S_v))$ for each subset S_v of S partitioned by A
20. return IG(S, A)

Pseudocode for the ID3 algorithm involves several steps, summarized as follows. First, initialize the tree by creating a root node. Next, check the base cases: if all examples in the dataset have the same label, return a single node with that label. If no attributes are left to use for splitting, return a node with the most common label in the dataset. Then, select the best Attribute based on the one with the highest information gain, where information gain is calculated by entropy, which formulas are as follows:

a. Entropy

$$H(S) = - \sum_{i=1}^k p_i \log_2(p_i) \quad (1)$$

The entropy $H(S)$ of dataset S is defined as k, the number of unique values in the target attribute T, and p_i , which is the proportion of S instances belonging to class i.

b. Information Gain

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \left(\frac{|S_v|}{|S|} \times H(S_v) \right) \quad (2)$$

The information gain $IG(S, A)$ of an attribute A based on values (A) are the unique values of attribute A, S_v is the subset of S where attribute A has values v, $|S_v|$ is the number of instances in S_v , and $|S|$ is the number of instances in the original dataset S.

Entropy measures the uncertainty or impurity in the dataset, and information gain represents the reduction in entropy after partitioning the dataset based on an attribute. After selecting the best Attribute, partition the dataset based

on that Attribute and recursively build the decision tree for each resulting subset. Finally, if any subset is empty after the split, create a leaf node with the most common label from the original dataset.

2.2 ACO (Ant Colony Optimization)

The ant colony algorithm is a distributed problem-solving technique inspired by ant behavior; it uses pheromone communication to develop solutions through interactions among ants to achieve an optimal result, where the pseudocode in the ACO is as follows:

Input: Problem instance, number of ants m , number of iterations max_iter , pheromone evaporation rate ρ , pheromone influence α , heuristic influence β

Output: Best solution found

1. Initialize pheromone levels τ on all edges to a constant value τ_0
2. for iteration = 1 to max_iter do
3. for ant $k = 1$ to m do
4. Construct a solution for ant k using the probabilistic transition rule
5. Evaluate the solution constructed by ant k
6. end for
7. Update pheromones using the solutions constructed by all ants
8. if a better solution is found, update the best solution
9. end for
10. return the best solution
- // Construct a solution for ant k
11. function ConstructSolution(k):
12. while the solution is not complete do
13. Select the next component based on the probabilistic transition rule
14. Add the selected component to the solution
15. end while
16. return the solution
- // Probabilistic transition rule
17. function TransitionProbability($\tau, \eta, \alpha, \beta$):
18. $P_{ij} = (\tau_{ij}^\alpha) * (\eta_{ij}^\beta) / \sum (\tau_{ik}^\alpha * \eta_{ik}^\beta)$ for all k in feasible components
19. return P_{ij}
- // Update pheromones
20. function UpdatePheromones($\tau, \text{solutions}, \rho$):
21. Evaporate pheromones: $\tau_{ij} = (1 - \rho) * \tau_{ij}$ for all edges (i, j)
22. for each solution in solutions do
23. for each edge (i, j) in the solution do
24. Deposit pheromones: $\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}$
25. end for
26. end for
- //pheromone deposit
27. function PheromoneDeposit(solution):
28. $\Delta\tau_{ij} = Q / L$, where Q is a constant and L is the length (cost) of the solution
29. return $\Delta\tau_{ij}$

Based on the pseudocode of ACO, the first step is to initialize the pheromone levels τ on all edges to an initial constant value τ_0 . Each algorithm iteration involves many ants m , constructing solutions based on a probabilistic transition rule. Each ant selects the next component in their route based on the current pheromone levels and heuristic information η . The transition probability P_{ij} is determined by the relative influence of these two factors, represented by α for the pheromone influence and β for the heuristic influence. The transition probability is calculated based on formulated, where N_i is the set of feasible components from node i as follows [25]:

$$P_{ij} = \frac{(\tau_{ij}^\alpha)(\eta_{ij}^\beta)}{\sum_{k \in N_i} (\tau_{ik}^\alpha)(\eta_{ik}^\beta)} \quad (3)$$

Once each ant has constructed a solution, then evaluate the quality. Subsequently, the pheromone levels are renewed in two main steps: evaporation and deposition. The evaporation process reduces the pheromone levels on all edges to prevent excessive accumulation, using the formula:

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (4)$$

Where ρ is the pheromone evaporation rate, following evaporation, a new pheromone is deposited on the edges, which is part of the solutions found by the ants with the amount of pheromone deposited refers to the formula where Q is a constant. L is the length (cost) of the solution.

$$\Delta\tau_{ij} = \frac{Q}{L} \quad (5)$$

This process is repeated for a predetermined number of iterations or until convergence is achieved. During these iterations, the best solution found is renewed continually. At the end of the iterations, the algorithm has the best optimum.

2.3 Combining ID3 and ACO

In this study, we combine ID3 and ACO to enhance the performance of decision tree construction for academic performance analysis. The integration of ACO aims to address the limitations of ID3, such as overfitting and computational inefficiencies. By leveraging ACO's optimization capabilities, we can optimize the structure of the decision tree, improve generalization to new data, and enhance the overall efficiency of the algorithm. Therefore, the steps we adopted in this study were as follows:

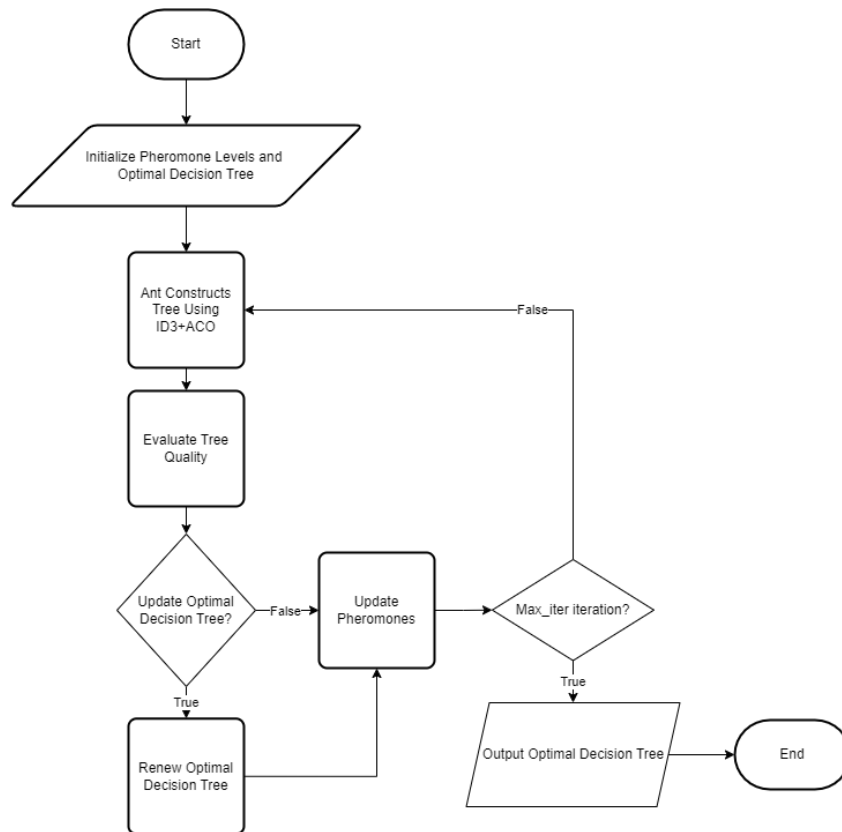


Figure 1. The Study Desain (ACO-ID3)

Based on Figure 1, we initialize the pheromone levels on all attributes and set the optimal decision tree to an initial state without any predefined structure. This step ensures that all attributes start with the same initial pheromone level and that no optimal tree exists beforehand. After the initialization, ants construct decision trees using the combined ID3 and ACO approach. Each ant builds a tree based on attributes selected using pheromone levels and ID3 rules. Once an ant constructs a tree, we evaluate its quality. This evaluation measures the tree's performance in terms of accuracy and generalization.

Subsequently, we check if the newly constructed tree is better than the current optimal decision tree. If the new tree performs better, the algorithm updates the optimal decision tree to this new tree. If not, the optimal decision tree remains unchanged. This step ensures that we always retain the best solution found so far. We then update the pheromone levels. It increases the pheromone on attributes used by flourishing trees and decreases the pheromone levels on other attributes. This process prevents overfitting and encourages the exploration of diverse solutions.

Afterward, we check whether it has reached the maximum number of iterations (max_iter). If it has reached the maximum iterations, the algorithm outputs the optimal decision tree found during the process. If not, we loop back to the step where ants construct new trees. The process repeats until we reach the maximum number of iterations or find an optimal solution. Finally, it outputs the optimal decision tree as the result and ends the process.

3. RESULT AND DISCUSSION

The sample data we obtained from the "Sistem Informasi Kemahasiswaan dan Akademik Universitas Indraprasta PGRI" website was as many as 91 instances. Some of these instances can be observed in Table 1:

Path 2	Final Exam ≤ 77.5 , Assignments ≤ 67.5 , Mid-Term Exam ≤ 45.0	Poor	Even with moderate final exam scores, low mid-term scores lead to poor performance.
Path 3	Final Exam ≤ 77.5 , Assignments ≤ 67.5 , Mid-Term Exam > 45.0 , Mid-Term Exam ≤ 69.0	Fair	Moderate mid-term scores improve the classification to fair.
Path 4	Final Exam ≤ 77.5 , Assignments ≤ 67.5 , Mid-Term Exam > 45.0 , Mid-Term Exam > 69.0	Good	Higher mid-term scores lead to a "good" classification.
Path 5	Final Exam ≤ 77.5 , Assignments > 67.5 , Mid-Term Exam ≤ 47.5	Fair	Better assignments and moderate mid-term scores lead to a "fair" classification.
Path 6	Final Exam ≤ 77.5 , Assignments > 67.5 , Mid-Term Exam > 47.5 , Mid-Term Exam ≤ 57.5	Fair	Consistent performance leads to a "fair" classification.
Path 7	Final Exam ≤ 77.5 , Assignments > 67.5 , Mid-Term Exam > 47.5 , Mid-Term Exam > 57.5 , Final Exam ≤ 62.5	Fair	Moderate final exam scores still result in a "fair" classification despite good performance in assignments and mid-terms.
Path 8	Final Exam ≤ 77.5 , Assignments > 67.5 , Mid-Term Exam > 47.5 , Mid-Term Exam > 57.5 , Final Exam > 62.5 , Assignments ≤ 82.5	Fair	Moderate assignment scores keep the classification fair despite higher final and mid-term scores.
Path 9	Final Exam ≤ 77.5 , Assignments > 67.5 , Mid-Term Exam > 47.5 , Mid-Term Exam > 57.5 , Final Exam > 62.5 , Assignments > 82.5	Good	Higher assignment scores lead to a "good" classification due to consistently high performance.
Path 10	Final Exam > 77.5 , Mid-Term Exam ≤ 72.5 , Final Exam ≤ 82.5	Good	High mid-term and moderate final exam scores result in a "good" classification.
Path 11	Final Exam > 77.5 , Mid-Term Exam ≤ 72.5 , Final Exam > 82.5	Excellent	High final exam scores and good mid-term scores lead to an excellent classification.
Path 12	Final Exam > 77.5 , Mid-Term Exam > 72.5	Excellent	Consistently high scores across all exams result in an excellent classification.

3.1 Evaluation in ACO-ID3 Algorithm with Hyperparameters

We evaluate the Model using cross-validation to ensure the robustness and reliability of the constructed decision trees. The process involves using the hyperparameters identified through grid search. The dataset is split into multiple folds, typically three in this implementation, where each fold is used once as a test set while the remaining folds serve as the training set. For each iteration, the decision tree is trained on the training set and evaluated on the test set, with the accuracy recorded. The overall model performance is determined by calculating the mean accuracy across all folds, and the variability in performance is assessed using the standard deviation of the accuracy scores. This method comprehensively evaluates the Model's effectiveness, ensuring that the selected hyperparameters lead to consistently high performance across different data subsets.

```
from sklearn.model_selection import KFold
# Cross-validation for ACO-ID3 with best parameters
kf = KFold(n_splits=3) # Reduced number of folds for faster computation
aco_id3_cv_scores = []
for train_index, test_index in kf.split(X):
    X_train_cv, X_test_cv = X.iloc[train_index], X.iloc[test_index]
    y_train_cv, y_test_cv = y.iloc[train_index], y.iloc[test_index]
    accuracy = aco_id3_with_params(X_train_cv, y_train_cv, X_test_cv, y_test_cv, best_params['max_iter'],
    best_params['num_ants'], best_params['alpha'], best_params['beta'], best_params['rho'])
    aco_id3_cv_scores.append(accuracy)
# Print cross-validation results
print(f"ACO-ID3 - Cross-Validation Accuracy with Best Params: {np.mean(aco_id3_cv_scores):.2f} (+/-
{np.std(aco_id3_cv_scores) * 2:.2f})")
```

3.1.1 Traditional ID3 vs ACO-ID3 Testing

The tests we carried out using cross-validation and comparing them with Traditional ID3 are depicted in Figure 3.

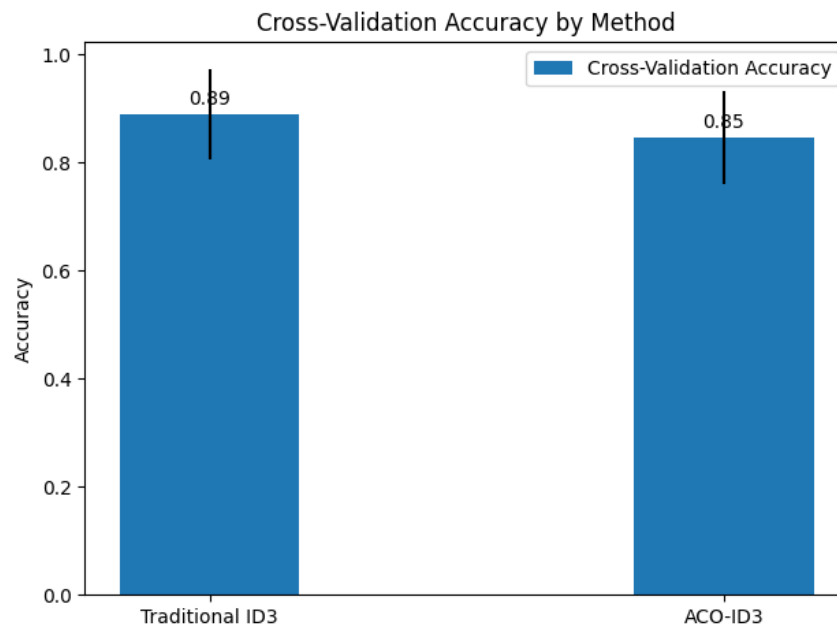


Figure 3. Cross-Validation Evaluation Model

Based on Figure 3, there are three points that we conclude, as follows:

- a. Cross-Validation Accuracy:
 1. Traditional ID3: The cross-validation accuracy of the Traditional ID3 model is 0.89, indicating that this Model correctly classifies approximately 89% of the instances on average across the cross-validation folds
 2. ACO-ID3: The cross-validation accuracy of the ACO-ID3 model is 0.85, showing that this Model correctly classifies around 85% of the instances on average across the folds.
- b. Error Bars:
 1. Traditional ID3: The error bar indicates a higher variability compared to the ACO-ID3 model, suggesting that the Traditional ID3 model's performance fluctuates more across different subsets of the data.
 2. ACO-ID3: The error bar shows a lower standard deviation, indicating more consistent performance across the folds.
- c. Comparative Analysis:
 1. Accuracy: The Traditional ID3 model shows a slightly higher average accuracy than the ACO-ID3 model. It suggests that, on average, the Traditional ID3 is more accurate in this dataset.
 2. Consistency: The ACO-ID3 model exhibits more consistent performance across different cross-validation folds, as the smaller error bar indicates.

Therefore, The Traditional ID3 model achieves a higher average accuracy (0.89) compared to the ACO-ID3 model (0.85), making it slightly better in terms of classification performance on this dataset. However, the ACO-ID3 model demonstrates more consistent performance with lower variability across cross-validation folds. While the ACO-ID3 algorithm aims to optimize feature selection using ant colony optimization, the Traditional ID3's straightforward approach seems to perform marginally better in this academic performance case.

3.2 Discussion

The constructed decision tree for academic performance analysis provides several critical insights into the determinants of student success in the Programming 3 course. Primarily, final exam scores emerge as the most significant predictor of overall academic performance. Students who achieve final exam scores above 77.5 are predominantly classified as "Good" or "Excellent," underscoring the importance of solid performance in final assessments for attaining higher classifications. In addition to final exam scores, mid-term exam results play a pivotal role, especially for students with lower final exam scores. High mid-term scores have the potential to elevate a student's classification from "Poor" to "Fair" or "Good," highlighting the necessity for consistent performance throughout the course rather than focusing solely on final exams. Although assignment scores are less influential than exam scores, they still contribute to the overall classification. High assignment scores can enhance a student's classification, particularly when complemented by good exam performance. These findings suggest that balanced and consistent effort across all types of assessments is essential for achieving the best possible academic performance classification.

The decision tree analysis provides valuable insights into targeted strategies for improving students' academic performance, particularly those classified as "Poor" or "Fair." Focused tutoring and support are crucial, especially in preparation for final exams, which significantly determine overall performance. Providing focused tutoring and review sessions leading up to the final exam can help students improve their understanding and retention of key concepts. Regular assessment and feedback throughout the semester can help identify areas where students struggle early,

allowing for timely interventions. Enhancing engagement with assignments through regular feedback and breaking down more significant assignments into smaller, more manageable tasks with regular deadlines can help students stay on track and avoid being overwhelmed. A balanced assessment approach incorporating various assessments such as projects, quizzes, and class participation alongside traditional exams can provide a more holistic evaluation of a student's capabilities and offer multiple opportunities for students to demonstrate their knowledge. Personalized learning plans based on the decision tree insights can address students' specific weaknesses, including tailored study schedules, targeted exercises, and additional resources focused on their areas of need. Academic mentoring and counseling services can also provide continuous support and guidance throughout the semester, helping students stay focused and motivated.

While constructing a decision tree, we found several differences that emerged when comparing the performance of the optimized ID3 algorithm using the Ant Colony Algorithm (ACO-ID3) with the traditional ID3 algorithm. The traditional ID3 algorithm achieves a higher average cross-validation accuracy of 0.89, indicating superior performance in correctly classifying instances. However, the ACO-ID3 algorithm, while slightly less accurate with an average accuracy of 0.85, demonstrates greater consistency and stability across different data subsets; this is evidenced by the lower variability in performance and smaller error bars associated with ACO-ID3, which suggests that it is more reliable in handling different portions of the dataset.

Based on this case, the traditional ID3 algorithm has several advantages, including its simplicity and ease of implementation and being computationally less intensive. However, it also has disadvantages, such as the tendency to overfit training data, especially if the dataset is small or noisy, and its approach might not lead to the globally optimal tree. On the other hand, the ACO-ID3 algorithm benefits from the optimization capabilities of ant colony optimization, which can lead to a more optimal decision tree by exploring various feature combinations. It also offers flexibility through fine-tuning with various hyperparameters to adapt to different datasets. Nevertheless, ACO-ID3 is more complex and computationally intensive, requiring more execution time due to its iterative nature and parameter tuning.

Therefore, future research should focus on optimizing the ID3 structure for academic performance analysis using the Ant Colony Algorithm by exploring various sets of hyperparameters to identify optimal settings that could improve the accuracy of the ACO-ID3 algorithm. Investigating hybrid models that combine ACO with other optimization techniques or machine learning algorithms could enhance the robustness and accuracy of the decision tree. Testing the ACO-ID3 algorithm on more extensive and diverse datasets would provide a more comprehensive evaluation of its generalizability and performance across different educational contexts. Additionally, incorporating features such as attendance records, participation in class activities, and extracurricular involvement could offer a more holistic view of academic performance. Developing real-time analysis tools using the ACO-ID3 algorithm could provide timely feedback and support for students, thereby enhancing their academic outcomes. By addressing these recommendations, future research can significantly advance the applicability and effectiveness of the ACO-ID3 algorithm in academic performance analysis, making it a valuable tool for educators and academic institutions.

4. CONCLUSION

Based on the study result, students' academic performance in the Programming 3 course reveals that final exam scores are the most critical indicator. Students with final exam scores above 77.5 typically achieve higher performance classifications, such as good or excellent. In contrast, those scoring 77.5 or below are likelier to be categorized as fair or poor, especially if their assignment scores are also low. Mid-term exam scores are crucial, particularly for students with lower final exam scores, as higher mid-term scores can improve their overall classification. Assignment scores, while influential, are less significant compared to exam scores. Higher assignment scores can enhance performance classifications but are less decisive than high final or mid-term exam scores. Thus, based on the research findings, the decision tree optimized with ACO is deemed suitable for use in educational settings for academic performance analysis. Although ACO-ID3 shows a slightly lower average accuracy than the traditional ID3, model evaluation indicates that ACO-ID3 demonstrates better consistency and stability with lower variability in performance across different data subsets. We suggest that ACO-ID3 still applies in educational contexts, particularly where consistent and reliable analysis is required. The method provides clear classifications and can assist in identifying key factors influencing student performance. By exploring various feature combinations and producing a more optimal decision tree structure, ACO-ID3 can be a tool to identify students needing additional support and develop more effective teaching strategies. Several steps have been obtained while constructing the decision tree of ACO-ID3. First, conducting experiments with various sets of hyperparameters to find the optimal configuration for ACO-ID3 can enhance the algorithm's accuracy and efficiency. Second, testing the algorithm on more extensive and diverse datasets will ensure its generalizability and consistent performance across different educational contexts. Third, developing hybrid models that combine ACO with other optimization techniques or machine learning algorithms can improve the robustness and accuracy of the decision tree. Fourth, incorporating additional features such as attendance records, participation in class activities, and extracurricular involvement can provide a more comprehensive analysis of academic performance. Finally, developing real-time analysis tools using ACO-ID3 can provide prompt feedback and support for students based on their ongoing academic performance. By implementing these recommendations, future

research can significantly enhance the effectiveness and applicability of the ACO-ID3 algorithm in academic performance analysis, making it a more reliable tool for educators and academic institutions.

REFERENCES

- [1] M. F. Rahman and F. Fadilah, "Klasifikasi Penerima Program Bantuan Beras Miskin Menggunakan Algoritma Iterative Dichotomiser 3," *Progresif: Jurnal Ilmiah Komputer*, vol. 18, no. 1, p. 101, Feb. 2022, doi: 10.35889/progresif.v18i1.797.
- [2] P. Chuenprasertsuk and K. Jearanaitanakij, "Improving the ID3 Algorithm By Filtering Out Attributes With Values Of 0 or 1," in *2022 6th International Conference on Information Technology (InCIT)*, 2022, pp. 173–176. doi: 10.1109/InCIT56086.2022.10067614.
- [3] I. Rasyid Munthe, S. Sarkum, and V. Sihombing, "Analysis Iterative algorithms Dichotomizer (ID3): The Satisfaction Study in Computer Laboratory," in *Proceedings of the Joint Workshop KO2PI and The 1st International Conference on Advance & Scientific Innovation, EAI*, 2018. doi: 10.4108/eai.23-4-2018.2277581.
- [4] I. Ida and M. Faisal, "Iterative Dichotomiser Three (ID3) Algorithm For Classification Community of Productive and Non-Productive," *JURNAL TEKNIK INFORMATIKA*, vol. 16, no. 1, pp. 80–88, May 2023, doi: 10.15408/jti.v16i1.28938.
- [5] A. Gusti and S. Hadianti, "Application of the C4.5 Algorithm to Predict the Effectiveness of Google Adwords Ads," *SISTEMASI*, vol. 12, no. 1, p. 21, Jan. 2023, doi: 10.32520/stmsi.v12i1.1978.
- [6] P. Chen, "Application of an Improved C4.5 Algorithm in Shopping Websites," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, IEEE, Jun. 2021, pp. 63–66. doi: 10.1109/ICAICA52286.2021.9498086.
- [7] C. Deng and Z. Ma, "Research on C4.5 Algorithm Optimization For User Churn," in *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*, IEEE, Aug. 2021, pp. 75–79. doi: 10.1109/CSAIEE54046.2021.9543367.
- [8] Y. Zheng, "Application of C4.5 algorithm in customer group classification of business websites," in *Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022)*, Y. Zhong, Ed., SPIE, Mar. 2023, p. 216. doi: 10.1117/12.2669803.
- [9] W. Wiguna and D. Riana, "Diagnosis of Coronavirus Disease 2019 (Covid-19) Surveillance Using C4.5 Algorithm," *Jurnal Pilar Nusa Mandiri*, vol. 16, no. 1, pp. 71–80, Mar. 2020, doi: 10.33480/pilar.v16i1.1293.
- [10] R. Tang and X. Zhang, "CART Decision Tree Combined with Boruta Feature Selection for Medical Data Classification," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, 2020, pp. 80–84. doi: 10.1109/ICBDA49040.2020.9101199.
- [11] M. Ozcan and S. Peker, "A classification and regression tree algorithm for heart disease modeling and prediction," *Healthcare Analytics*, vol. 3, p. 100130, Nov. 2023, doi: 10.1016/j.health.2022.100130.
- [12] L. Villalobos-Arias, C. Quesada-López, A. Martínez, and M. Jenkins, "Hyper-Parameter Tuning of Classification and Regression Trees for Software Effort Estimation," in *Trends and Applications in Information Systems and Technologies*, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. Ramalho Correia, Eds., Cham: Springer International Publishing, 2021, pp. 589–598.
- [13] S. M. Teki, B. Banothu, and M. K. Varma, "An Un-realized Algorithm for Effective Privacy Preservation Using Classification and Regression Trees," *Revue d'Intelligence Artificielle*, vol. 33, no. 4, pp. 313–319, Oct. 2019, doi: 10.18280/ria.330408.
- [14] T. Daniya, M. Geetha, and K. Suresh Kumar, "Classification And Regression Trees With Gini Index," *Advances in Mathematics: Scientific Journal*, vol. 9, no. 10, pp. 8237–8247, Sep. 2020, doi: 10.37418/amsj.9.10.53.
- [15] L. Ju, L. Huang, and S.-B. Tsai, "Online Data Migration Model and ID3 Algorithm in Sports Competition Action Data Mining Application," *Wirel Commun Mob Comput*, vol. 2021, pp. 1–11, Jul. 2021, doi: 10.1155/2021/7443676.
- [16] M. O. Okwu and L. K. Tartibu, "Ant Colony Algorithm," in *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, M. O. Okwu and L. K. Tartibu, Eds., Cham: Springer International Publishing, 2021, pp. 33–41. doi: 10.1007/978-3-030-61111-8_4.
- [17] S. Fidanova, "Ant Colony Optimization," in *Ant Colony Optimization and Applications*, S. Fidanova, Ed., Cham: Springer International Publishing, 2021, pp. 3–8. doi: 10.1007/978-3-030-67380-2_2.
- [18] F. Zhang, "Ant Colony Algorithm for Distributed Constrained Optimization," in *2023 2nd International Conference for Innovation in Technology (INOCON)*, 2023, pp. 1–5. doi: 10.1109/INOCON57975.2023.10101321.
- [19] S. Biswas, S. A. Nusrat, and N. Tasnim, "Grid-Based Pathfinding Using Ant Colony Optimization Algorithm," in *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems*, A. B. Reddy, S. Nagini, V. E. Balas, and K. S. Raju, Eds., Singapore: Springer Nature Singapore, 2023, pp. 259–269.
- [20] F. Zhang, "Application of ant colony algorithm in distributed artificial intelligence," in *2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs)*, 2022, pp. 76–80. doi: 10.1109/AIoTCs58181.2022.00107.
- [21] C.-W. Tsai and M.-C. Chiang, "Ant colony optimization," in *Handbook of Metaheuristic Algorithms*, Elsevier, 2023, pp. 139–161. doi: 10.1016/B978-0-44-319108-4.00021-6.
- [22] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "An Ant Colony Algorithm for Classification Rule Discovery," in *Data Mining*, IGI Global, 2020. doi: 10.4018/9781930708259.ch010.
- [23] A. K. Nugroho, I. Permadi, Y. I. Kurniawan, A. Hanifa, and Nofiyati, "Decision tree using ant colony for classification of health data," in *AIP Conference Proceedings*, 2023, p. 020002. doi: 10.1063/5.0128787.
- [24] B. Yan and S. Danning, "Teaching Quality Evaluation and Software Implementation Based on ID3 Decision Tree Algorithm," in *2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE)*, 2022, pp. 337–341. doi: 10.1109/ISAIEE57420.2022.00076.
- [25] N. Selvia, E. W. Ambarsari, and N. Dwitianti, "Korelasi Gejala Penyakit Flu Pada Anak Balita Dengan Menggunakan Algoritma Semut," *JITEK*, vol. 2, no. 2, pp. 167–174, 2022.